International Journal of Automation and Power Engineering, 2012, 1: 108-111
Published Online July 2012

www.ijape.org

The Design of RFID Tag for "Mywallet"

Xianghao Nan
Information Security Research Group, School of Electronics Engineering and Computer Science, Peking University
Email: Nanxh2001@163.com

(**Abstract**) In recently years, tags of RFID have been developed rapidly[1][2][3]. Different tags may have different applications but the security requirements are of the same, i.e.: anti-reproducing and anti-forgery. With respect to reproducing, it can only be solved physically. And with respect to anti-forgery, it can only be solved logically. "Mywallet "is designed to meet the need of dual direction authentication without CPU.

Keywords: RFID Tag; CPK Cryptosystem; Mywallet; Dual Direction Authentication**.**

1. Technical Requirements

1.1 Two Kinds of Authentication Concept

The design characteristic of Mifare reflects different understanding to the authentication relationship. Among Writer, tag, Reader, Mifare emphasizes on mutual authentication between Reader and tag. Thus, it must offer tag certain "intelligent" function. Accordingly, simple dynamic devices such as cipher machine and random number generator are set in tag, to barely conduct interactive authentication with Reader. Such interactive authentication cannot be equal, because Reader is an intelligent device, and tag is a memory device. This causes irreconcilable fatal conflict. Practice proves that authentication and encryption of Mifare is not reliable. Cracking of smart card Mifare and the appearance of simulated decipher ghost are the recent examples[4][5][6].

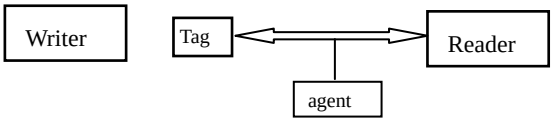

Fig1 Relationship 1 among Writer, Reader and Tag

This concept is needed to enhance the level of intelligence of tag. Low intelligence is easy to cause security issue.
The other authentication concept is to emphasize mutual authentication between Writer and Reader among the three (i.e., Writer, tag, and Reader), with tag only as agent for Writer. Both Writer and Reader are active and intelligent devices, and the mutual authentication can be equal. Thus, it greatly improves authentication security, and reduces strict requirements to tag. CPK-based identity authentication technology can be directly applied to mutual authentication of Writer and Reader and to provide digital signature, verification, data encryption and decryption.

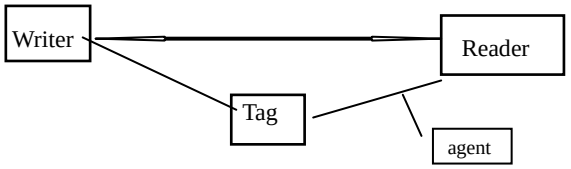

Fig2 Relationship 2 among Writer, Reader, Tag

1.2 Two Kinds of Authentication Networks

The authentication network is constructed among the Readers, Writers and tags. All Readers, Writers and tags can verify each other.

1) Centralized Authentication Network

The centralized authentication network is formed between Readers and tags, and Writers and tags [7] as Fig 3.

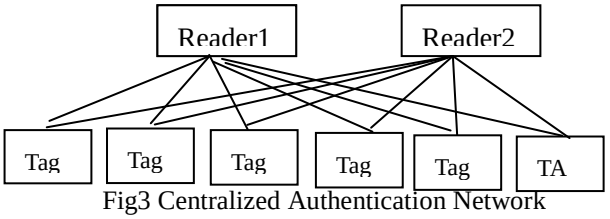

Fig3 Centralized Authentication Network

2) Horizontal Authentication Network

The horizontal authentication network is formed between Writers and Writers, Readers and Readers, Writers and Readers as Fig4.

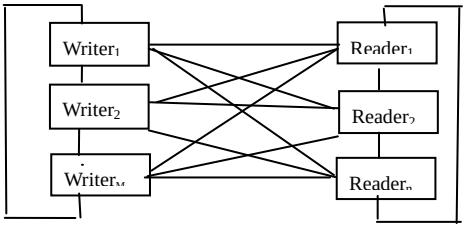

Fig4 Horizontal Authentication Network

1.3 Two Kinds of Business Requirements

Digital Signature: the signature is done in the Readers and Writers. The length of sign code must be very short.

Data Encryption: the encryption and decryption is done in the Readers and Writers. The length of data to be encrypted or decrypted must not be limited, and the length of data must not be enlarged after encryption.

2. System Structure

The electronic wallet used in Bus traffic system can reflect the common business requirements. Therefore, we are going to take the bus card for example to illustrate the design principle of tag.

2.1 Key Distribution

The CPK key distribution protocol is different from traditional methods such as PKI and so on[8][9]. The key is generated by KMC and distributed to Manufacturers, Enterprises, Writers and Readers. Writers are dispersed over selling points, and Readers are placed at every bus. The distribution of key is a kind of authorization. Key configuration is as follows:

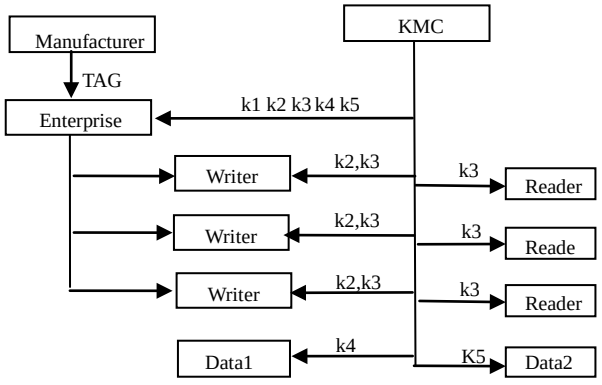

Fig 5 Configuration of authentication keys

Manufacturer: Defines UID for every tag and supplies enterprises;

Enterprise: The operator of enterprise holds CPK-card having his private key and symmetric keys k1, k2, k3, k4, k5. The operator signs on UID and defines m for every tag, and writes m into EEPROM on tag or connects the circuits in accordance with m, and encrypts m: $E_{ki}(m_i)=n_i$, and writes n_i to TAG to supply selling points;

Writer: The operator of Writer holds CPK-card having his private key and symmetric keys k2, k3. The operator manages the deposit and balance in tag. The operator is in charge of deposit by signing and encrypting;

Reader: The ticket seller holds CPK-card only having his private key and symmetric key k3. The seller manages the balance by signing and encrypting.

Enterprise ID-card contents are as below:

1	Z1: Authentication Para	16B	$\mathbf{E}_{PWD}(R1)=Z1$
2	Z2: Authentication Para	16B	$\mathbf{E}_{R1}(R_1)\oplus R1=Z2$
3	Identity definition	25B	EnterpriseID,
4	private key	32B	$\mathbf{E}_{R1}$(enterprise)＝Y
5	Spare key k1	48B	$\mathbf{E}_{R1}(k1)=W_1$ for enterprise
6	Spare key k2	48B	$\mathbf{E}_{R1}(k2)=W_2$ for Writer
7	Spare key k3	48B	$\mathbf{E}_{R1}(k3)=W_3$ for Reader
7	Spare key k4	48B	$\mathbf{E}_{R1}(k4)=W_4$ for dynamic data
7	Spare key k5	48B	$\mathbf{E}_{R1}(k5)=W_5$ for static data
8	Issue unit	25B	KMC
9	Signature of issue unit	48B	SIG_{kmc} (MAC)

2.2 Data Structure

Data structure is composed of items and sections. The structure is as follows:

	1. UID	2. deposit	3. Balance	4.Data(dyn)	5.data(static)
Item 1	FSR_1(plain) (4B)	FSR_2(plain) (4B)	FSR_3(plain) (4B)	FSR_4(plain) (4B)	FSR_5(plain) (4B)
Item 2	$Code_1=E_1$(UID) (4B)	$Code_2=E_2(data_2)$ (4B)	$Code_3=E_3(data_3)$ (4B)	$Code_4=E_4(data_4)(8B\times i)$	$Code_5=E_5(data_5)(8B\times i)$
Item 3	$Signer_1$ (20B)	$Signer_2$ (20B)	$Signer_3$ (20B)	$Signer_4$ (20B)	$Signer_5$ (20B)
Item 4	$Sign_1$(32B)	$Sign_2$(32B)	$Sign_3$(32B)	$Sign_4$(32B)	$Sign_5$(32B)
Item 5	$n_1=Ek_1(m_1)$(5B)	$n_2=Ek_2(m_2)$ (5B)	$n_3=Ek_3(m_3)$ (5B)	$n_4=Ek_4(m_4)$ (5B)	$n=Ek_5(m_5)$ (5B)

Section includes UID section, deposit section, balance section, dynamic data section and static data section. The size of data section will be decided by the need of user and the design of data section must be matched with file managing system.

Item 1 is the initial state of FSR in plain form. The data length is equal to the toggle number of FSR.

Item 2 is the coded data in the section. The data length is the multiple of 4 Byte.

Item 3 is the signer's identity. The data length is fixed to 20 Bytes.

Item 4 is the sign code. The code length is fixed to 32 Bytes.

Item 5 is the coded parameter of m. the data length is 5 Bytes.

2.3 Controller Structure

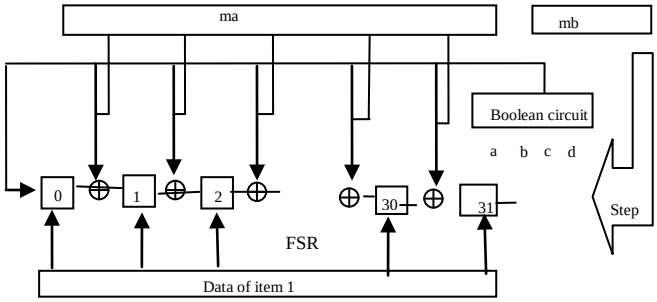

Fig 6 TAG structure of Controller

The controller is composed of 32-toggled feedback shift register FSR with 31 mod-2 adders, 8-toggled stepping unit and a Boolean circuit, as shown in Fig 6.

The Boolean circuit is composed of 4-gate combinatorial circuit:

$$F(x) = a\ \underline{b}\ \underline{d} + \underline{a}\ b\ \underline{d} + b\ c\ d + \underline{b}\ \underline{c}\ d$$

The input a, b, c, d comes from any 4 outputs of the 32-toggled FSR.

The feedback circuit of the FSR is decided by the given parameter m, and m is divided into ma and mb. ma is a 31-bit random number, and controls the 31 mod-2 adders respectively, 1 stands for connected, 0 stands for disconnected. mb is an 8-bit random number, and controls the shifting steps of the FSR. The contents of ma and mb can never be all zeros. m can be stored in EEPROM of tag or directly changed into physical circuit. The encrypted form of m is n. n is used for sending. The Writer or Reader must decrypt n first, $D_k(n)=m$, and then decides the feedback relation and the number of shifting steps. Every sector can have separated m.

3. Protocol Design

3.1 Authentication Protocol

In this scheme, the authentication protocol is based on CPK cryptosystem and truth logic [10][11][12]. The challenge and response between the verifying side (Writer or Reader) and tag is as follows.

Here, deposit process will be taken as an example to describe the operation protocol.

1) The verifying side reads out the plain FSR_i in item 1 and writes into the control register.

Also read out n_i in item 6 decrypts n_i:

$$D_{ki}(n_i)=m_i$$

and divides m_i into ma_i and mb, and decide the feedback relation according to ma_i and the stepping number according to mb_i.

The tag takes out the plain initial value from item 1 in data structure and writes it into its FSRi. Thus, the two sides have the same state. It is called ‘state 0’.

2) The FSR_i of both sides shifts mb_i steps, and the state is changed into ‘state 1’, and the first challenge and response is carried out in this state.

The verifying side: if the content of 0^{th}-toggle of FSR_i is 0, then send out the 8-bit contents of 1^{st} - 8^{th} toggles of FSR_i, or send out the 8-bit contents of 16^{th} -23rd toggles of FSR_i.

The tag side: if the content of 0^{th}-toggle of FSR_i is 0, then check the content of 1^{st} - 8^{th} toggles of FSR_i, or check the content of 16^{th} -23rd toggles of FSR_i. If it is correct, then set the flag on 1, and the next procedure continues, or set the flag on 0, and the process is terminated, so as to prevent the “middle-man” and “signal-copying” attacks. The probability of error occurs is 1/512.

3) The FSR_i of both sides shift mb_i steps, and enters into “state 2”, and the second challenge and response is carried out in this state.

The tag side: if the content of 0^{th}-toggle of FSR_i is 0, then check the content of 1^{st} - 8^{th} toggles of FSR_i, or check the content of 16^{th}-23^{rd} toggles of FSR_i.

The verifying side: if the content of 0^{th}-toggle of FSR_i is 0, then check the content of 1^{st} -8^{th} toggles of FSR_i, or check the content of 16^{th}-23^{rd} toggles of FSR_i. If it is correct, then set the flag on 1, and the next procedure continues, or set the flag on 0, and the process is terminated, so as to prevent the “middle-man” and “signal-copying” attacks. The probability of error occurs is 1/512.

Now the task of tag is completed, the following steps are processed only in the verifying sides (Writer or Reader)

3.2 Decryption and Verification Protocol

The FSR_i of verifying side shifts mb_i steps, and turns into “state 3”. The decryption and verification are processed in this state. Because the operation is only processed in the verifying side, the schemes of encryption and signature may be selected at one’s will.

Decryption: two schemes may be provided.

The first scheme is to take the 4-bytes of “state 3” as a random number (*RN*), and to add it to the data. Suppose that the length of data is a multiple of 4 Byte, then

For j:=0 to n-1 do begin $RN \oplus code_j = data_j$;

FSR_i shifts mb_i steps;

end;

The second scheme is to use block cipher. The block length may be 4 Bytes or 8 Bytes.

For j:=0 to n-1 do $D_{ki}(code_j) = data_j$;

Verification: *SIGNER* is the public key of the signer in the following function.

$$VER_{SIGNER}(data_i, s_i)=c_i' \ (i=1..5)$$

If $c_i=c_i'$, then the flag is set on 1, and turned into next step, or the flag is set on 0, and the process is terminated.

The data in the sections of UID and static data would not be changed, and now the work is ended.

3.3 Encryption and Signature Protocol

Both of the control registers shift mb_i steps, and enter into 'state 4'.

In verifying side, 'state 4' of FSR_i is encrypted with k*i* to create a new 'state 0' of FSR_i

$$E_{ki}(\text{plain 'state 4'}) = \text{new 'state 0'}$$

The new 'state 0' is written into control register of verifying side and sent to tag.

In tag side, if the flag of tag is flag=1, then the new 'state 0' is accepted and written into the item 1 in data structure. Or it will be denied.

The FSR_i of verifying side shifts mb_i steps and turns into new 'state 1'. The state will be used in the first authentication for the next operation.

The FSR_i of verifying side shifts mb_i steps and turns into new 'state 2'. The state will be used in the second authentication for the next operation.

The FSR_i of verifying side shifts mb_i steps and turns into new 'state 3'. The operation of signature and encryption will be processed in this new 'state 3' for the verification and decryption in the next operation.

Signature: *signer* is private key in the following signature function:

$$SIG_{signer}(data_i) = (s_i, c_i) \quad (i=1..5)$$

The $sign_i=(s_i,c_i)$ is sent to tag, If the flag is flag=1, then the sign is accepted, and written into the tag or it will be denied.

Encryption: two schemes may be provided.

The first scheme is to take the 4-Byte of "state 3" as a random number (*RN*), and to add it to the code. Suppose that the length of code is a multiple of 4 Byte, then

For *j*:=0 to *n*-1 do begin $RN \oplus data_j = code_j$;

FSR_i shifts mb_i steps;

end;

The second scheme is to use block cipher. The block length is 4 Bytes or 8 Bytes.

For *j*:=0 to *n*-1 do $E_{ki}(data_j) = code_j$;

The $code_j$ is sent to tag. If the flag of tag is flag=1, then the $code_j$ is accepted, or it will be denied.

Summary

This system closely connects CPK authentication system with Writer and Reader, to jointly protect tag security. This solution directs the difficulties of dissection analysis and simulation analysis to the difficulties of cryptography analysis in CPK-card. With respect to tag design, since it has 2^{39} different structures, and is hardly to find the same structured tag.

The signals that are exchanged between the verification side and tag are sent plainly but are used secretly. Therefore, the signal copying attack is meaningless. The FSR is shifting by different steps, and provides no successive sequence to be used in cryptographic analysis.

There are two kinds of requirements in transactions. The first requirement is that the tag is not needed to have signature function. The signature may be done by the verification device such as Writer or Reader, just like the small-amount payment system. Tag is only a non-intelligent storage tool without signature function, and it cannot provide the evidence of responsibility. However, it can provide the evidence of the authenticity of tag itself and can check the authenticity of verification side. The second requirement is that the tag must have the signature functions. The signature must be done by tag itself. In such a case, the tag must be intelligent, and can provide the evidence of responsibility, such as CPK-card.

REFERENCES

[1] Gerhard de Koning Gans Radboud University Nijmegen RFID Security, Black Hat Sessions: Part VI - March 26, 2008.

[2] Greg Pote, APSCA, Current Situation and Future Perspective of Smart Card Business in Asia Pacific Japan PKI Forum International Conference Tokyo, 24 February 2005.

[3] Department of Defense Biometrics, Common Access Card (CAC) Biometric Integration Overview, www.biometrics.dod.mil.

[4] Flavio D. Garcia, Gerhard de Koning Gans, et el, Dismantling MIFARE Classic, Institute for Computing and Information Sciences, Radboud University Nijmegen.

[5] Roel Verdult, Security analysis of RFID tags, June 25, 2008

[6] Bart Jacobs & the Mifare Team, Smart Cards in Public Transport: the Mifare Classic Case,Jacobs – opening, EIPSI, Eindhoven 22/4/'08 – p.1/.

[7] Stephen Wilson, Embedded PKI- The Emerging Stage of the Art, Asia PKI Forum 6th International Symposium, Chengdu, China.

[8] NIST, A Synopsis of Federal Information Processing Standard (FIPS) 201 for Personal Identity Verification (PIV) of Federal Employees and Contractors, March 2005.

[9] Adi Shamir, SQUASH – A New MAC with Provable Security Properties for Highly Constrained Devices Such as RFID Tags, Computer Science department, The Weizmann Institute.

[10] Vincent Naessens, RFID Security and Privacy A Research Survey, Studiedag Rabbit Project.

[11] Xianghao Nan, Identity Authentication – Technical Basis of Cyber Security, China Industry and Electronics Publishing House, Jun 2011.

[12] Xianghao Nan, CPK Cryptosystem (v7.0), Computer Security, Sep 2011.

Author Introduction

Xianghao Nan

Dean, South China Information Security Institute

Doctorial Tutor, Information Engineering University, PLA

Part-time Professor, Institute of Computer Science and Technology, Peking University